Worked Examples using R, *Introductory Statistics, 7th ed.* by Neil Weiss

Charles Carter*

October 1, 2014

Contents

*cccarter@troy.edu

Abstract

This paper consists of the worked examples in each chapter of *Introductory Statistics, 7th Edition* by Neil Weiss, using the R programming language

1 The Nature of Statistics

1.1 Descriptive Statistics

This example contains no code.

1.2 Inferential Statistics

This example contains no code.

1.3 Classifying Statistical Studies

This example contains no code.

1.4 Classifying Statistical Studies

This example contains no code.

1.5 Simple Random Samples

```
#create vector of officials
library(prob)

## Error in library(prob):  there is no package called 'prob'

off <- c('G', 'L', 'S', 'A', 'T')
#part a, list of samples of size 2
urnsamples(off, 2)

## Error in eval(expr, envir, enclos):  could not find function
"urnsamples"

#part d, list of samples of size 4
urnsamples(off, 4)

## Error in eval(expr, envir, enclos):  could not find function
"urnsamples"
```

1.6 Random-Number Tables

```
#generate 15 random integers between 1 and 728
sample(1:728, 15)

##  [1] 279 419 151 207 654 655 253  53 455 644 183 570 420 339 521
```

1.7 Systematic Random Sampling

```
#declare variables
pop <- 728
sos <- 15
division <- floor(pop / sos)
division

## [1] 48

start <- sample(1:division, 1)
start

## [1] 9

#generate sequence
s <- seq(start, pop, division)
s

##  [1]   9  57 105 153 201 249 297 345 393 441 489 537 585 633 681
```

1.8

1.9

1.10

1.11

1.12

1.13

2 Organizing Data

2.1 Variables and Data

This example contains no code.

2.2 Variables and Data

This example contains no code.

2.3 Variables and Data

This example contains no code.

2.4 Variables and Data

This example contains no code.

2.5 Grouping Quantitative Data

```r
#read in the data
invest <- read.csv("data/Tb02-01.txt")
str(invest)
```

```
## 'data.frame': 40 obs. of  1 variable:
##  $ DAYS: int  70 64 99 55 64 89 87 65 62 38 ...
```

```r
w <- cut(invest$DAYS, c(30, 40, 50, 60, 70, 80, 90, 100), right = FALSE)
invest$CAT <- w
table(invest$CAT)
```

```
## 
##  [30,40)  [40,50)  [50,60)  [60,70)  [70,80)  [80,90) [90,100) 
##        3        1        8       10        7        7        4
```

```r
x <- table(invest$CAT)
y <- prop.table(x)
z <-  merge(x, y, by.x = "Var1", by.y = "Var1")
z
```

```
##       Var1 Freq.x Freq.y
## 1  [30,40)      3  0.075
## 2  [40,50)      1  0.025
## 3  [50,60)      8  0.200
## 4  [60,70)     10  0.250
## 5  [70,80)      7  0.175
## 6  [80,90)      7  0.175
## 7 [90,100)      4  0.100
```

2.6

2.7

2.8

2.9

2.10

2.11

2.12

2.13

2.14

2.15

2.16

2.17

2.18

3 Descriptive Measures

4 Probability Concepts

5 Discrete Random Variables

6 The Normal Distribution

6.1 Example 6.1,

6.2 Example 6.2,

6.3 Example 6.3,

6.4 Example 6.4,

6.5 Example 6.5,

6.6 Example 6.6,

6.7 Example 6.7,

6.8 Example 6.8,

6.9 Example 6.9,

6.10 Example 6.10,

6.11 Example 6.11,

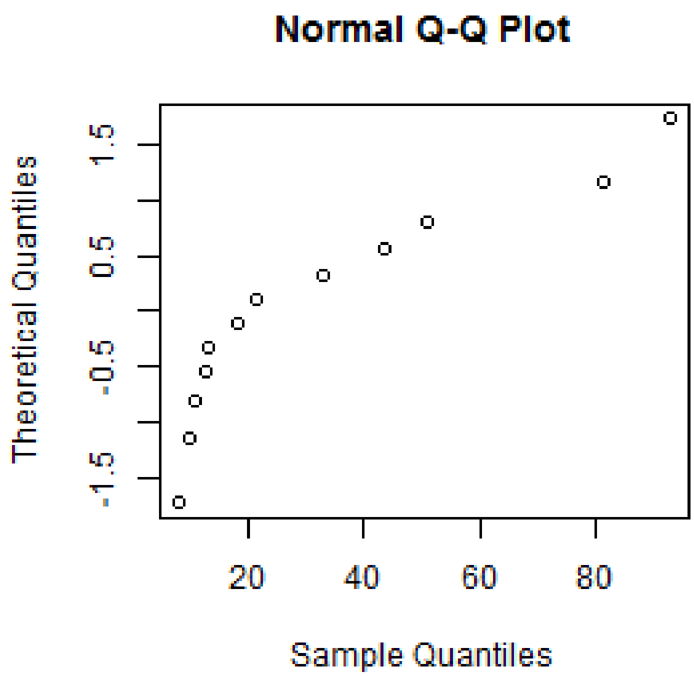
6.12 Example 6.12,

6.13 Example 6.13, Using Technology to Obtain Normal Percentiles

```
mu <- 100
sigma <- 16
ptile <- qnorm(0.90, mu, sigma)
ptile

## [1] 120.5048
```

6.14 Example 6.14, Normal Probability Plots

```
#read in the data
income <- read.csv("data/Tb06-03.txt")
str(income)

## 'data.frame': 12 obs. of  1 variable:
##  $ AGI: num  9.7 93.1 33 21.2 81.4 51.1 43.5 10.6 12.8 7.8 ...

qqnorm(income$AGI, datax = TRUE)
```



7 The Sampling Distribution of the Sample Mean

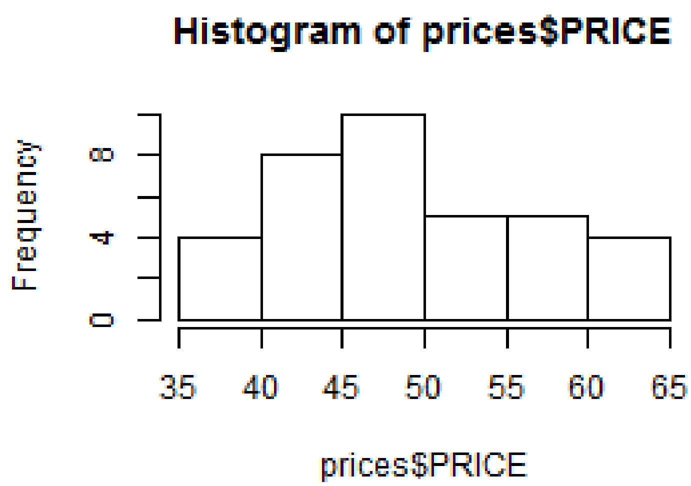
8 Confidence Intervals for One Population Mean

8.1 Example 8.1, Estimating a Population Mean

```
#read in the data
prices <- read.csv("data/Tb08-01.txt")
str(prices)
```

```
## 'data.frame': 36 obs. of  1 variable:
##  $ PRICE: num  53.8 54.4 45.2 42.9 49.9 48.2 41.6 58.9 48.6 53.1 ...
```

```
hist(prices$PRICE, breaks = 5)
```




```
sum <- sum(prices$PRICE)
n <- nrow(prices)
mu <- sum / n
#alternatively
mu1 <- mean(prices$PRICE)
```

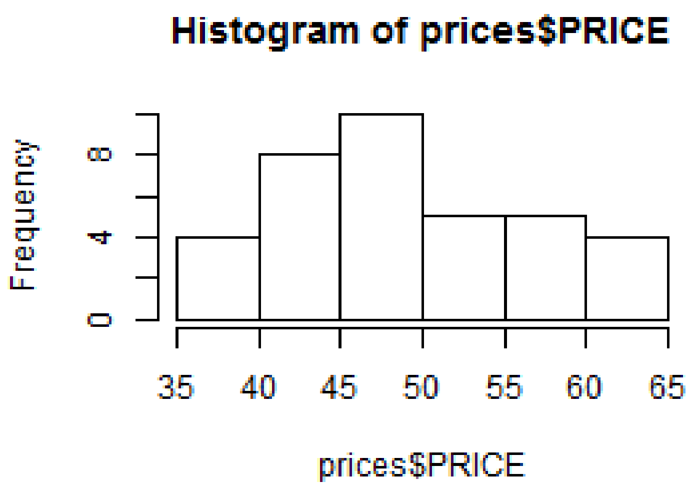
The estimated population, μ, from the sample mean, $\bar{x}$, is 49.2777778.

8.2 Example 8.2, Introducing Confidence Intervals

```r
#read in the data
prices <- read.csv("data/Tb08-01.txt")
str(prices)
```

```
## 'data.frame': 36 obs. of  1 variable:
##  $ PRICE: num  53.8 54.4 45.2 42.9 49.9 48.2 41.6 58.9 48.6 53.1 ...
```

```r
hist(prices$PRICE, breaks = 5)
```




```r
sum <- sum(prices$PRICE)
n <- nrow(prices)
mu <- sum / n
sigma <- 7.2
s <- sigma / sqrt(n)
cat(sum, n, mu, sigma, s)
```

```
## 1774 36 49.27778 7.2 1.2
```

```r
confidence.interval <- simple.z.test(prices$PRICE, sigma, conf.level = 0.9544)
```

```
## Error in eval(expr, envir, enclos):  could not find function
"simple.z.test"
```

```r
confidence.interval
```

```
## [1] 46.87889 51.67667
```

8.3 Example 8.3, Interpreting Confidence Intervals

8.4 Example 8.4, The One-Sample z-Interval Procedure

8.5 Example 8.5, Using Technology to Obtain a z-Interval

8.6 Example 8.6, Introducing the Margin of Error

8.7 Example 8.7, Sample Size for Estimating μ

8.8 Example 8.8, Finding the t-Value Having a Specified Area to the Right

8.9 Example 8.9, The One-Sample t-Interval Procedure

8.10 Example 8.10, The One-Sample t-Interval Procedure

8.11 Example 8.11, Choosing a Confidence Interval Procedure

9 Hypothesis Tests for One Population Mean

9.1 Example 9.1, Choosing the Null and Alternative Hypotheses

This example contains no code.

9.2 Example 9.2, Choosing the Null and Alternative Hypotheses

This example contains no code.

9.3 Example 9.3, Choosing the Null and Alternative Hypotheses

This example contains no code.

9.4 Example 9.4, The Logic of Hypothesis Testing

Null hypothesis is $H_0 : \mu = 454$.
Alternative hypothesis is $H_\alpha : \mu \neq 454$.

```
# load the data file
weights <- read.csv("data/Tb09-01.txt")
str(weights)

## 'data.frame': 25 obs. of  1 variable:
##  $ WEIGHT: int  465 456 438 454 447 449 442 449 446 447 ...

#declare and initialize variables
mu <- 454
sigma <- 7.8
n <- 25
xbar <- mean(weights$WEIGHT)
ztest <- (xbar - mu) / (sigma / sqrt(n))
ztest

## [1] -2.564103

#determine the result of the test
result <- pnorm(ztest)
result

## [1] 0.005172149

#use simple z test from UsingR package
library(UsingR)

## Error in library(UsingR): there is no package called 'UsingR'

conf.int <- simple.z.test(weights$WEIGHT, sigma = sigma, conf.level = 0.9544)

## Error in eval(expr, envir, enclos):  could not find function
"simple.z.test"

conf.int

## [1] 446.8814 453.1186
```

The claimed weight of the population is μ per bag, 454 grams. The mean sample weight is $\bar{x}$ per bag, 450 grams. The z value is -2.5641026, which is more than two standard deviations below the population mean.

9.5 Example 9.5, Type I and Type II Errors

This example contains no code.

9.6 Example 9.6, Obtaining the Critical Values

```
left.tail <- qnorm(0.05)
left.tail
```

```
## [1] -1.644854
```

```
right.tail <- qnorm(0.95)
right.tail
```

```
## [1] 1.644854
```

```
two.tail.left <- qnorm(0.025)
two.tail.left
```

```
## [1] -1.959964
```

```
two.tail.right <- qnorm(0.975)
two.tail.right
```

```
## [1] 1.959964
```

9.7 Example 9.7, The One-Sample z-Test

Null hypothesis is $H_0 : \mu = \$51.46$.
Alternative hypothesis is $H_\alpha :> \$51.46$

```
# load the data file
books <- read.csv("data/Tb09-05.txt")
str(books)
```

```
## 'data.frame': 40 obs. of  1 variable:
##  $ PRICE: num  56 46.2 47.3 54 53.7 ...
```

```
#declare and initialize variables
mu <- 51.46
sigma <- 7.61
n <- 40
xbar <- mean(books$PRICE)
right.tail = 0.01
ztest <- (xbar - mu) / (sigma / sqrt(n))
ztest
```

```
## [1] 2.850829
```

```
right.crit <- qnorm(1 - right.tail)
right.crit
```

```
## [1] 2.326348
```

The z statistic is 2.8508286, which is greater than the critical value of 2.3263479, so we reject the null hypothesis.

9.8 Example 9.8, The One-Sample z-Test

Null hypothesis is $H_0 : \mu = 800$.
Alternative hypothesis is $H_\alpha :< 800$

```
# load the data file
rda <- read.csv("data/Tb09-06.txt")
str(rda)
```

```
## 'data.frame': 18 obs. of  1 variable:
##  $ CALCI: int  686 433 743 647 734 641 993 620 574 634 ...
```

```
#declare and initialize variables
mu <- 800
sigma <- 188
n <- 18
xbar <- mean(rda$CALCI)
left.tail = 0.05
ztest <- (xbar - mu) / (sigma / sqrt(n))
ztest
```

```
## [1] -1.187287
```

```
left.crit <- qnorm(left.tail)
left.crit
```

```
## [1] -1.644854
```

The z statistic is -1.1872874, which is less than the critical value of -1.6448536, so we do not reject the null hypothesis.

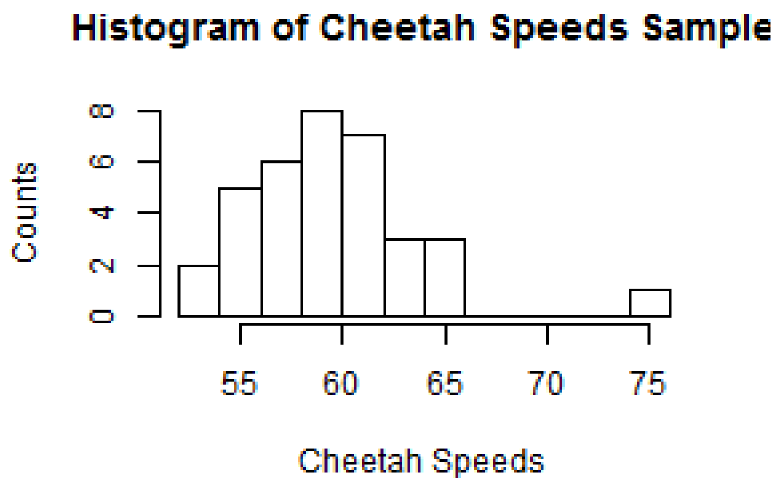
9.9 Example 9.9, The One-Sample z-Test

Null hypothesis is $H_0 : \mu = 60$.
Alternative hypothesis is $H_\alpha :\neq 60$

```
# load the data file
cheetah <- read.csv("data/Tb09-07.txt")
str(cheetah)
```

```
## 'data.frame': 35 obs. of  1 variable:
##  $ SPEEDS: num  57.3 57.5 59 56.5 61.3 57.6 59.2 65 60.1 59.7 ...
```

```
#histogram of the data set
hist(cheetah$SPEEDS, breaks = 15, xlab = "Cheetah Speeds", ylab = "Counts", main = "Histogram of
```




```r
#declare and initialize variables
mu <- 60
sigma <- 3.2
n <- 35
xbar <- mean(cheetah$SPEEDS)
tails = 0.05
ztest <- (xbar - mu) / (sigma / sqrt(n))
ztest

## [1] -0.8768475

crits <- qnorm(c( tails / 2, (1 - tails / 2)))
crits

## [1] -1.959964  1.959964
```

The z statistic is -0.8768475, which is less than the critical values of -1.959964, 1.959964, so we do not reject the null hypothesis.

9.10 Example 9.10,

9.11 Example 9.11,

9.12 Example 9.12,

9.13 Example 9.13,

9.14 Example 9.14,

9.15 Example 9.15,

9.16 Example 9.16,

9.17 Example 9.17,

9.18 Example 9.18,

9.19 Example 9.19,

9.20 Example 9.20,

9.21 Example 9.21,

10 Inferences for Two Population Means

11 Inferences for Population Standard Deviations

12 Inferences for Population Proportions

13 Chi-Square Procedures

14 Descriptive Methods in Regression and Correlation

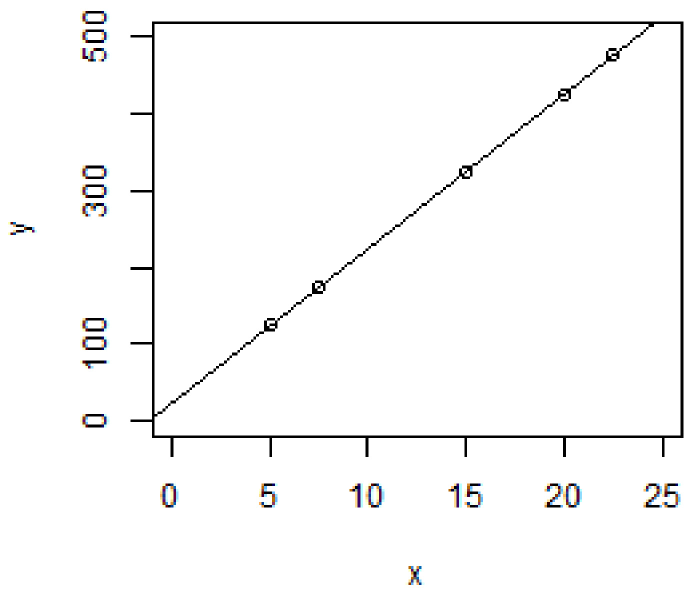
14.1 Linear Equations

```
# load the data file
wp <- read.csv("data/Tb14-01.txt", sep = "\t")
str(wp)
```

```
## 'data.frame':  5 obs. of  2 variables:
##  $ TIME: num  5 7.5 15 20 22.5
##  $ COST: int  125 175 325 425 475
```

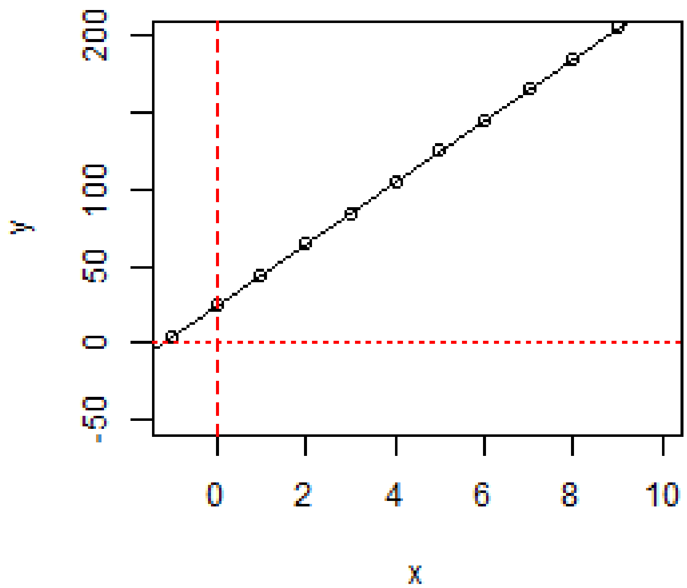
```
x <- wp$TIME
y <- 25 + (20 * x)
line <- lm(y ~ x)
```

```
plot(x, y, xlim = c(0,25), ylim = c(0,500))
abline(line)
```



14.2 y-Intercept and Slope

```
# load dummy data
x <- c(-1:10)
y <- 25 + (20 * x)
line <- lm(y ~ x)
plot(x, y, xlim = c(-1,10), ylim = c(-50,200))
abline(line)
abline(v=0,col="red", lty = 2)
abline(h=0,col="red", lty = 3)
```



14.3 Introducing the Least Squares Criterion

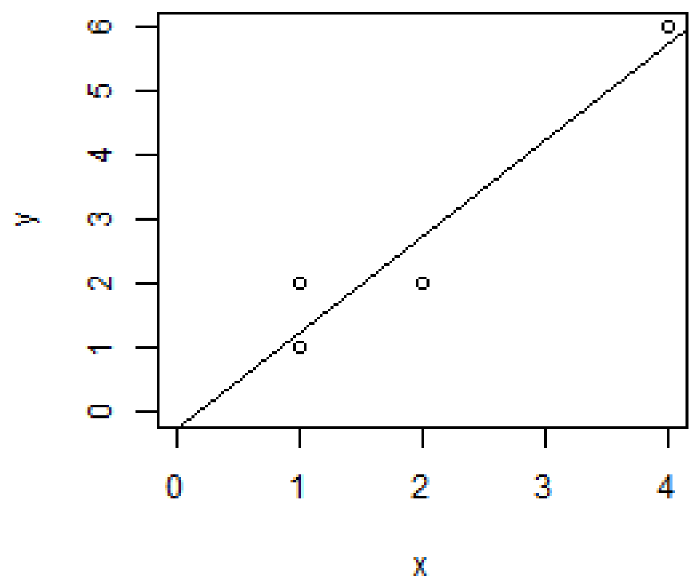
```
x <- c(1,1,2,4)
y <- c(1,2,2,6)
df <- data.frame(x,y)
df
```

```
##   x y
## 1 1 1
## 2 1 2
## 3 2 2
## 4 4 6
```

```
plot(df, xlim = c(0, 4), ylim = c(0, 6))
line <- lm(y ~ x)
line
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Coefficients:
## (Intercept)            x
##       -0.25         1.50
```

```r
abline(line)
```



14.4 The Regression Equation

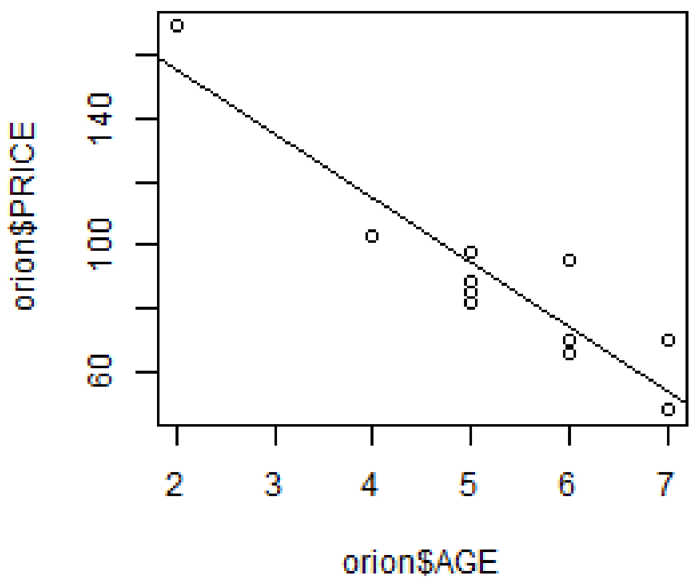
```r
# load Orion data
orion <- read.csv("data/Tb14-02.txt", sep = "\t")
str(orion)
```

```
## 'data.frame': 11 obs. of  2 variables:
##  $ AGE  : int  5 4 6 5 5 5 6 6 2 7 ...
##  $ PRICE: int  85 103 70 82 89 98 66 95 169 70 ...
```

```r
s <- lm(orion$PRICE ~ orion$AGE)
s
```

```
##
## Call:
## lm(formula = orion$PRICE ~ orion$AGE)
##
## Coefficients:
## (Intercept)    orion$AGE
##      195.47       -20.26
```

```r
plot(orion$AGE, orion$PRICE)
abline(s)
```




```
# find y for x equals 3 and 4
s$coefficients
```

```
## (Intercept)   orion$AGE
##   195.46847   -20.26126
```

```
intercept <- s$coefficients[[1]]
slope <- s$coefficients[[2]]
three.year.old.Orion <- intercept + ( slope * 3)
three.year.old.Orion
```

```
## [1] 134.6847
```

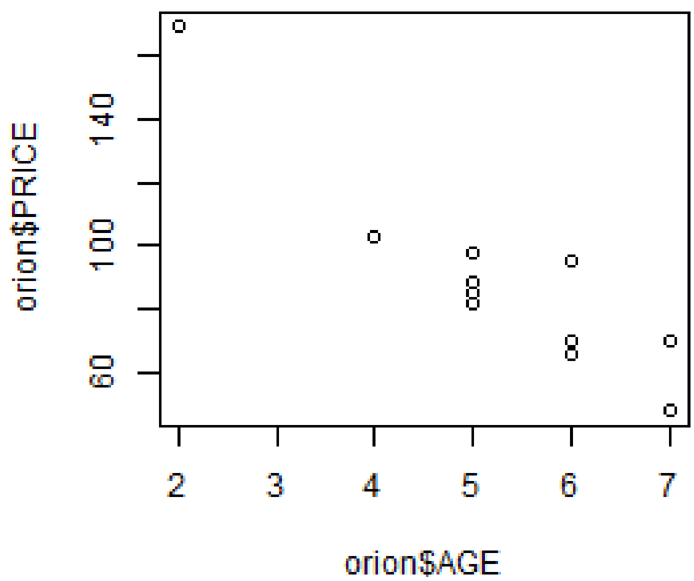
```
four.year.old.Orion <- intercept + ( slope * 4)
four.year.old.Orion
```

```
## [1] 114.4234
```

14.5 Using Technology to Obtain a Scatter Diagram

```
# load Orion data
orion <- read.csv("data/Tb14-02.txt", sep = "\t")
str(orion)
```

```
## 'data.frame': 11 obs. of  2 variables:
##  $ AGE  : int  5 4 6 5 5 5 6 6 2 7 ...
##  $ PRICE: int  85 103 70 82 89 98 66 95 169 70 ...

plot(orion$AGE, orion$PRICE)
```

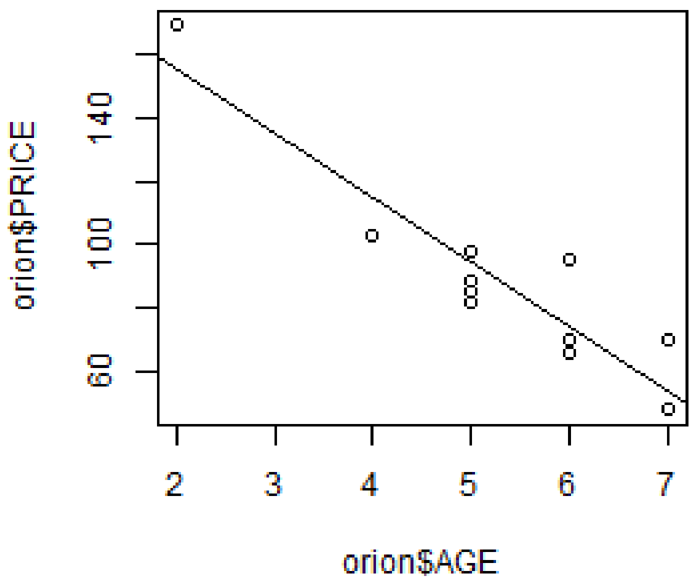


14.6 Using Technology to Obtain a Regression Line

```
# load Orion data
orion <- read.csv("data/Tb14-02.txt", sep = "\t")
str(orion)

## 'data.frame': 11 obs. of  2 variables:
##  $ AGE  : int  5 4 6 5 5 5 6 6 2 7 ...
##  $ PRICE: int  85 103 70 82 89 98 66 95 169 70 ...

s <- lm(orion$PRICE ~ orion$AGE)
s

##
## Call:
## lm(formula = orion$PRICE ~ orion$AGE)
```

```
##
## Coefficients:
## (Intercept)    orion$AGE
##      195.47       -20.26
```

```
plot(orion$AGE, orion$PRICE)
abline(s)
```



14.7 Introduces the Coefficient of Determination

```
# load Orion data
orion <- read.csv("data/Tb14-02.txt", sep = "\t")
str(orion)
```

```
## 'data.frame': 11 obs. of  2 variables:
##  $ AGE  : int  5 4 6 5 5 5 6 6 2 7 ...
##  $ PRICE: int  85 103 70 82 89 98 66 95 169 70 ...
```

```
y.sub.ybar <- orion$PRICE - mean(orion$PRICE)
y.sub.ybar
```

```
##  [1]  -3.6363636  14.3636364 -18.6363636  -6.6363636   0.3636364
##  [6]   9.3636364 -22.6363636   6.3636364  80.3636364 -18.6363636
## [11] -40.6363636
```

```
y.sub.ybar.sqr <- y.sub.ybar^2
y.sub.ybar.sqr
```

```
##  [1]   13.2231405  206.3140496  347.3140496   44.0413223    0.1322314
##  [6]   87.6776860  512.4049587   40.4958678 6458.3140496  347.3140496
## [11] 1651.3140496
```

```
t <- data.frame(orion,y.sub.ybar,y.sub.ybar.sqr)
s <- lm(orion$PRICE ~ orion$AGE)
yhat <- s$coefficients[[1]] + (s$coefficients[[2]] * t$AGE)
t$yhat <- yhat
yhat.sub.ybar <- t$yhat - mean(t$PRICE)
t$yhat.sub.ybar <- yhat.sub.ybar
y.sub.yhat.sqr <- (t$PRICE - t$yhat)^2
t$y.sub.yhat.sqr <- y.sub.yhat.sqr
yhat.sub.ybar.sqr <- (yhat - mean(t$PRICE))^2
t$yhat.sub.ybar.sqr <- yhat.sub.ybar.sqr
t
```

```
##    AGE PRICE  y.sub.ybar y.sub.ybar.sqr      yhat yhat.sub.ybar
## 1    5    85  -3.6363636     13.2231405  94.16216      5.525799
## 2    4   103  14.3636364    206.3140496 114.42342     25.787060
## 3    6    70 -18.6363636    347.3140496  73.90090    -14.735463
## 4    5    82  -6.6363636     44.0413223  94.16216      5.525799
## 5    5    89   0.3636364      0.1322314  94.16216      5.525799
## 6    5    98   9.3636364     87.6776860  94.16216      5.525799
## 7    6    66 -22.6363636    512.4049587  73.90090    -14.735463
## 8    6    95   6.3636364     40.4958678  73.90090    -14.735463
## 9    2   169  80.3636364   6458.3140496 154.94595     66.309582
## 10   7    70 -18.6363636    347.3140496  53.63964    -34.996724
## 11   7    48 -40.6363636   1651.3140496  53.63964    -34.996724
##    y.sub.yhat.sqr yhat.sub.ybar.sqr
## 1        83.94522          30.53445
## 2       130.49460         664.97245
## 3        15.21703         217.13386
## 4       147.91819          30.53445
## 5        26.64792          30.53445
## 6        14.72900          30.53445
## 7        62.42424         217.13386
## 8       445.17198         217.13386
## 9       197.51644        4396.96071
## 10      267.66139        1224.77069
## 11       31.80554        1224.77069
```

```
sst <- sum(t$y.sub.ybar.sqr)
sst
```

```
## [1] 9708.545
```

```
ssr <- sum((t$yhat - mean(t$PRICE))^2)
ssr
```

```
## [1] 8285.014
```

```
r.sqrd <- ssr / sst
r.sqrd

## [1] 0.8533733

sse <- sum((t$PRICE - t$yhat)^2)
sse

## [1] 1423.532
```

14.8 Computing Formulas for the Sum of Squares

```
# load Orion data
orion <- read.csv("data/Tb14-02.txt", sep = "\t")
str(orion)

## 'data.frame': 11 obs. of  2 variables:
##  $ AGE  : int  5 4 6 5 5 5 6 6 2 7 ...
##  $ PRICE: int  85 103 70 82 89 98 66 95 169 70 ...

xx <- orion$AGE^2
yy <- orion$PRICE^2
xy <- orion$AGE * orion$PRICE
t <- data.frame(orion, xx, yy, xy)
t

##    AGE PRICE xx    yy  xy
## 1    5    85 25  7225 425
## 2    4   103 16 10609 412
## 3    6    70 36  4900 420
## 4    5    82 25  6724 410
## 5    5    89 25  7921 445
## 6    5    98 25  9604 490
## 7    6    66 36  4356 396
## 8    6    95 36  9025 570
## 9    2   169  4 28561 338
## 10   7    70 49  4900 490
## 11   7    48 49  2304 336

sum.x <- sum(t$AGE)
sum.y <- sum(t$PRICE)
sum.xx <- sum(t$xx)
sum.yy <- sum(t$yy)
sum.xy <- sum(t$xy)
sst <- sum.yy - (sum.y^2 / nrow(t))
sst

## [1] 9708.545

num <- (sum.xy - (sum.x * sum.y / nrow(t)))^2
den <- sum.xx - sum.x^2 / nrow(t)
ssr <- num / den
ssr
```

```
## [1] 8285.014
sse <- sst - ssr
sse
## [1] 1423.532
```

14.9 Using Technology to Obtain a Coefficient of Determination

```
# read in the data
orion <- read.csv("data/Tb14-02.txt", sep = "\t")
str(orion)
## 'data.frame': 11 obs. of  2 variables:
##  $ AGE  : int  5 4 6 5 5 5 6 6 2 7 ...
##  $ PRICE: int  85 103 70 82 89 98 66 95 169 70 ...
s <- lm(orion$PRICE ~ orion$AGE)
summary(s)
##
## Call:
## lm(formula = orion$PRICE ~ orion$AGE)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.162  -8.531  -5.162   8.946  21.099
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   195.47      15.24  12.826 4.36e-07 ***
## orion$AGE     -20.26       2.80  -7.237 4.88e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.58 on 9 degrees of freedom
## Multiple R-squared:  0.8534,Adjusted R-squared:  0.8371
## F-statistic: 52.38 on 1 and 9 DF,  p-value: 4.882e-05
summary(s)$r.squared
## [1] 0.8533733
```

14.10 The Linear Correlation Coefficienbt

```
# read in the data
orion <- read.csv("data/Tb14-02.txt", sep = "\t")
str(orion)
```

```
## 'data.frame': 11 obs. of  2 variables:
##  $ AGE  : int  5 4 6 5 5 5 6 6 2 7 ...
##  $ PRICE: int  85 103 70 82 89 98 66 95 169 70 ...
```

```r
# initialize n, x, and y
n <- nrow(orion)
x <- orion$AGE
y <- orion$PRICE
# plug into formula
num <- sum(x * y) - (sum(x) * sum(y) / n)
den <- sqrt((sum(x^2) - (sum(x)^2/n))  *  (sum(y^2) - (sum(y)^2 / n)))
r <- num / den
r
```

```
## [1] -0.9237821
```

14.11 Using Technology to Obtain a Linear Correlatiion Coeffecient

```r
# read in the data
orion <- read.csv("data/Tb14-02.txt", sep = "\t")
str(orion)
```

```
## 'data.frame': 11 obs. of  2 variables:
##  $ AGE  : int  5 4 6 5 5 5 6 6 2 7 ...
##  $ PRICE: int  85 103 70 82 89 98 66 95 169 70 ...
```

```r
cor(orion)
```

```
##              AGE      PRICE
## AGE    1.0000000 -0.9237821
## PRICE -0.9237821  1.0000000
```

15 Inferential Methods in Regression and Correlation

16 Analysis of Variance (ANOVA)